Student Failures: A Knowledge Discovery Application

Agapito Ledezma and Daniel Borrajo
ledezma@inf.uc3m.es, dborrajo@ia.uc3m.es

Abstract-- Knowledge Discovery in Databases (KDD) is a field that has increased its importance during the last years. Large volumes of data from very different areas that require exhaustive analysis to find relations between its attributes have caused the increase of interest on this field. KDD has been used in domains of science and in applied domains, like marketing, stock market, etc. In this paper we have focused on applying KDD in a specific domain, the Curricular Plan in Computer Science, to find the relationships that give us the explanation about the graduation ratio for students.

***Index terms--* Knowledge Discovery in Databases, Knowledge acquisition, Data Mining.**

I. INTRODUCTION

In recent years, the growth in the number of databases and the increase of their size create the need to apply new methods and techniques that help humans to extract useful information from them. *Knowledge discovery* is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [1]. People in different kinds of fields are experiencing this growth on the data that they manipulate. For this reason, the number of tools available has increased too.

University Education is currently under revision in many countries. For instance, the Spanish government has recently launched a quality analysis plan for improving teaching, research and management of Spanish universities. There are many factors that can be analyzed within teaching. Given the complex relationships that might affect teaching activities, we need more powerful tools and techniques.

In this paper we applied the KDD techniques to one specific example of such analysis: knowledge acquisition about the relationship between the Curricular Plan for a Short Degree in Computer Science and the student graduation.

We have used two learning systems to carry out such analysis: C4.5 [6] that generated rules; and Naive Bayes [8] that generated conditional probabilities about the domain.

The paper is organized as follows. A general description of the Knowledge Discovery in Databases process and the domain of application are described in Section II. The experimental setup is described in Section III. Finally we present our conclusions and the future work.

II. BACKGROUND

A. The KDD process

The term KDD was coined in the KDD workshop in 1989 [4] to highlight that the *knowledge* is the final product of a data-driven discovery. In agreement with Fayyad [5], the KDD process is interactive and iterative and involves numerous steps with many decisions being made by the user. The steps involved in the KDD process are: defining the goal of the application, creating the target data set, data cleaning and preprocessing, data reduction and projection, matching the goal of KDD process to a particular data mining method, choosing the data mining algorithm, data mining, interpreting mined patterns and consolidating discovery knowledge.

As we saw, one step inside the KDD process is the *data mining*, which involves a repeated iterative application of a machine learning algorithm. In this work we have used two learning schemes for the classification task.

C4.5 is an algorithm derived from the simple ID3 [2] divide-and-conquer algorithm for producing decision trees. A decision tree (Figure 1) is a representation of the relation between a conclusion-decision and the attributes about a domain.

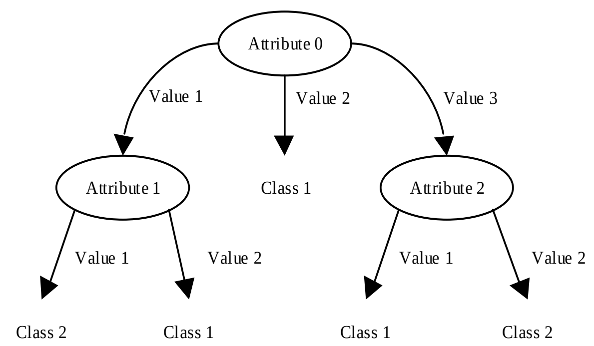

Figure 1. An example of decision tree

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Avd. de la Universidad Nº 30
28911 Leganés - Madrid, Spain

The first attribute used for splitting the data set is known as the *root*. Each division is a branch that corresponds to the value of an attribute and the final node for each branch is a leaf.

In a creates the decision tree:

1. Each node is an attribute and each branch from this node is a possible value of this attribute. A leaf of the tree is the expected class for a example. The explanation of the classification of an instance is the route from the root to the leaf.
2. Each node is associated with an attribute still unused in the route from the root.
3. To know what is the attribute that is more significant to split, the algorithm uses the entropy. When the entropy is lower, the uncertainty is less and the attribute is more significant.

The decision tree is generated from a data set and it represents the whole set of examples. In some cases the tree can be very big and complex. To simplify the tree C4.5 uses *pruning,* that is, replace one part of the tree (subtree) for a single leaf. This procedure is performed when the expected error in the subtree is higher than the expected error of the leaf used to replace it.

A disadvantage of decision trees is that despite the pruning it can still be cumbersome and complex, hence very difficult to understand by humans. One way to solve this problem is to rewrite the tree as a collection of rules. When we have the decision tree, C4.5 rules generator obtains a set of rules from the tree. Each rule is a correspondence of the path from the root to the leaf. Moreover the rule generator removes some irrelevant conditions.

Another classifier used in this work is the Naive Bayes algorithm. This algorithm treats all attributes as completely independent and with equal importance and classifies a new example in accordance with the class with higher probability given the attributes' values. Naive Bayes generates the hypothesis computing the frequency of occurrences.

B. *The Learning Task*

The task we wanted to study is: given that we have a curricular plan for a given subject at a particular University, how does the structure and the composition of the curricular plan influence the student's academic behavior? To find an answer to this question we posed this problem like a learning task on which we applied the KDD process. More concretely, the learning task consisted on given the curricula of students, described in term of grades, predict if they are going to leave the university or not. This task has similarly posed by Tom Mitchell [3].

In our university we have a short degree in Computer Science whose curricula is composed of a set of courses of different types. The curricula is based on fulfilling a number of credits where each credit is equivalent to 10 hours of class. The courses can be core, obligatory, optional and courses of free election. Core courses refer to courses that are in all Spanish universities curricula in the same degree. Obligatory courses are demanded by each university. Optional courses are offered by each short degree and university and the student can choose from several to complete a number of credits. Free election courses are offered by the entire university including other short degrees.

To finish the Short Degree in Computer Science, students have to obtain a total of 211 credits of which 168 must be the sum of core and obligatory courses, 20 credits by optional courses and 23 credits by free election courses.

Additional to these 211 credits, students have to obtain six credits by humanity courses and three credits of English courses. When the students have the total credits they must carry out a Final Degree Project.

The Curricular Plan is composed by three academic years divided in two four-month periods each one. Each academic year has about ten courses.

The first academic year, students have to pass at least two courses. Moreover they have to pass in two consecutive academic years, at least the 65% of the total credit of the first year of the curricular plan (around 42). Also students have three consecutive academic years to pass the first year of the curricular plan. Finally, students have six examination sessions to pass each course.

III. Experimental Setup

We made two experiments with different data sets, obtaining different results.

A. *First Experiment*

Following the KDD process steps, once we defined the goal of our study, we selected the data set for our application. The original data was supplied in text format and with irrelevant information, and we had to apply filters to obtain the relevant information for our work. After the preprocessing of the data set we had a set with 107 instances (student's records), 75 of which are of the positive class or *graduated* (students that had finished the short degree) and 32 are of the negative class or *failed* (students that had not concluded the short degree). The curricular plan of the Short Degree in Computer Science in our work had 33 courses of which the first 20 were chosen for our experiment because only those had a significant influence on the prediction. Each instance had two attributes per course (40 attributes per instance): the grade obtained in the course and the number of the "examination session" in which the student had obtained the last grade. The possible values for the first attribute were "not taught", "not presented", "failed", "passed", "notable", "good", "excellent", "annulled", "comparable" and "recognized". The possible values for the later attribute were 0 to 6 and "unnecessary". The value "0" was used to indicate that the course was "not taught" and the value "unnecessary" was to point out that course is "comparable" or "recognized". We used only two attributes per course because we dealt with "inconclusive" data due to the nature of the original data.

Once the data was selected and preformatted, we used a supervised learning system that built a decision tree from examples, C4.5. C4.5 constructed a set of rules (if-then rules) from decision tree to make the output more comprehensible. Figure 2 shows an example of the generated rules[1].

C4.5 also predicts the percentage of unseen cases in which the classification made by the rule will be correct in some percentage (shown in square brackets) of unseen cases.

```
Rule 1:
Course_19_session = 0
        ->  class failed  [93.3%]

Rule 9:
Course_12_session = 0
        ->  class failed  [92.2%]

Rule 3:
Course_18_grade = not_presented
        ->  class failed  [84.1%]

Rule 7:
Course_7_session = 6
        ->  class failed  [70.7%]

Rule 12:
Course_12_session = unnecessary
        ->  class graduated  [93.3%]

Rule 8:
Course_18_grade = notable
        ->  class graduated  [93.0%]

Rule 4:
Course_18_grade = passed
Course_19_session = 1
        ->  class graduated  [91.4%]

Rule 11:
Course_12_session = 3
        ->  class graduated  [73.1%]
```

Figure 2 Rules generated by C4.5

The interpretation of those rules is the following:

Rule 1: students that not coursed the course 19 did not conclude the short degree. The reason for that is that the student did not reach the course because s/he had been expelled before.
Rule 9: same as rule 1.
Rule 3: if a student obtained a "not presented" as the final grade of the course 18, the student failed, because the student was expelled before.
Rule 7: students that obtained the grade of course 7 in the sixth examination session failed. Therefore, this is a key course.
Rule 12: students that have the course 12 "compared" or "recognized" had finished the Short Degree.
Rule 8: students have obtained "notable" in course 18 concluded the Short Degree.
Rule 4: the students passed course 18 and passed course 19 in the first examination session have a high probability of concluding the Short Degree.
Rule 11: students that passed the course 12 in session 3 had finished the Short Degree.

The rules generated by C4.5 gave us information that is possible to infer by just analyzing the data set. This might be caused because we dealt with data that do not contain some attributes which may be essential to for the domain representation. The absence of these attributes may make it impossible to discover significant knowledge about the domain. Moreover the number of examples was small. These reasons took us to the second experiment.

B. Second Experiment

Like in the first experiment, we used the first 20 courses of the curricular plan of the Short Degree in Computer Science. This time we had more attributes per course. We used 120 attributes per instance (one for each examination session of the course).
In this experiment we had more information about the courses, but we had less instances to learn from (62). The possible values of all attributes were "not taught", "not presented", "failed", "passed", "notable", "good", "excellent", "comparable", "recognized" and "unfinished". "Unfinished" referred to the student that had attended at least one examination session and "unnecessary" referred to the student that had passed the course in a previous examination session. Once we pre-processed the data, we ran C4.5 on the data set and obtained similar rules to the ones of the first experiment. This output made us think that the last courses of the curricular plan did not influence the students' graduation. Therefore, we eliminated the lasts courses attributes to generate another set of rules that would give us a good prediction. We obtained a new set of rules (Figure 3) that permitted us to find some relations among the courses. However, they were not yet suitable to be used for generating any general conclusion about the domain.

```
Processing tree 0

Final rules from tree 0:

Rule 10:
        Course_16_s6 = not_taught
        ->  class failed  [95.3%]

Rule 4:
        Course_7_s3 = failed
        ->  class failed  [91.2%]

Rule 9:
        Course_13_s4 = unfinished
        ->  class failed  [79.4%]

Rule 1:
        Course_2_s5 = failed
        ->  class failed  [70.7%]

Rule 2:
        Course_2_s5 = passed
        ->  class failed  [70.7%]

Rule 7:
        Course_2_s5 = unnecessary
        Course_7_s3 = unnecessary
        Course_13_s4 = unnecessary
        ->  class graduated  [83.1%]

Default class: failed
```

Figure 3 Rules generated by C4.5

The meaning of the rules is as follows:
Rule 10: students that "not taught" the course 16 did not conclude the short degree. That is because the student had been expelled before studying this course.
Rule 4: students that obtained a "failed" in session three of course 7 did not conclude the Short Degree. Therefore, it seems that it is a key course.
Rule 9: students that did not finish the course 13 did not conclude the Short Degree.

[1] In order not to make public the names of specific courses, we have removed their actual names.

Rule 1: students that did not pass the course 2 before session 5 have a high probability of not concluding the Short Degree.
Rule 2: students that passed the course 2 in session 5, have a high probability of not concluding the Short Degree.
Rule 7: students that pass the course 2 before session five and course 7 before session three and the course 13 before session 4 have a high probability of finishing the Short Degree.

In order to obtain a more probabilistic knowledge that would allow us to formulate more conditional conclusions, we applied a Naive-Bayes algorithm to the data set. One difficulty associated to the Naive-Bayes algorithm was that the output of this algorithm was difficult to interpret because it is numeric (probabilistic output). We used the implementation of the Naive Bayes algorithm in Mooney's tool [14] to generate the following probabilities table (Figure 4).

```
Output:
(        (0.3508772         0.64912283)

    (    (0.4      0.5135135)         (0.45     0.13513513)
         (0.15     0.0)      (0.0      0.0)      (0.0
         0.0)      (0.0      0.0)      (0.0
         0.027027028)        (0.0      0.0)      (0.0
         0.0)      (0.0      0.3243243)          (0.0
         0.0))

    (    (0.2      0.2972973)          (0.15
         0.054054055)        (0.0      0.0)      (0.0
         0.0)      (0.0      0.0)      (0.0      0.0)
         (0.0      0.0)      (0.6      0.16216215)
         (0.0      0.0)      (0.05     0.4864865)
         (0.0      0.0))
```

Figure 4. Naive Bayes output.

To make the output easier to understand we generated a set of charts about various aspects of the output. This confirmed us, as it was expected, that the courses of the first year in the curricular plan had a major influence in the graduation of the students. To extend this conclusion, we generated a series of graphs that represented the courses in the first year of the short degree.

As an example of our analysis, we show the charts generated for Course 1. In the first chart (Figure 1) we can observe that in most cases the students that have graduated have passed the first course before examination session 3. Also, with respect to failures, we can see that about a 6% of students that had not concluded the short degree have been expelled for this course.

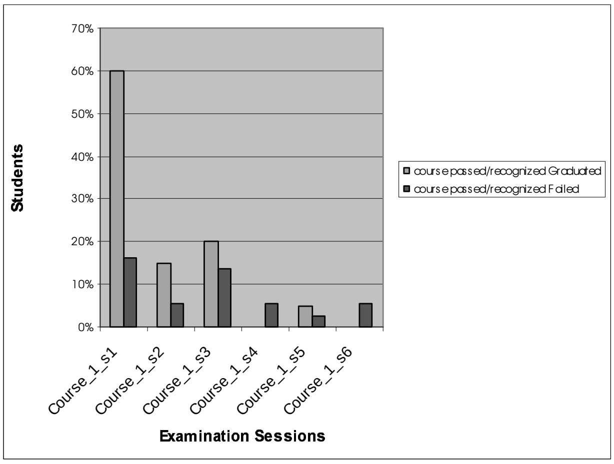

Figure 5. Course 1. Passed course.

In the next chart (Figure 6) we confirmed the previous premise. That is, students that have graduated have passed the course 1 in the examination session 5 at the very latest.

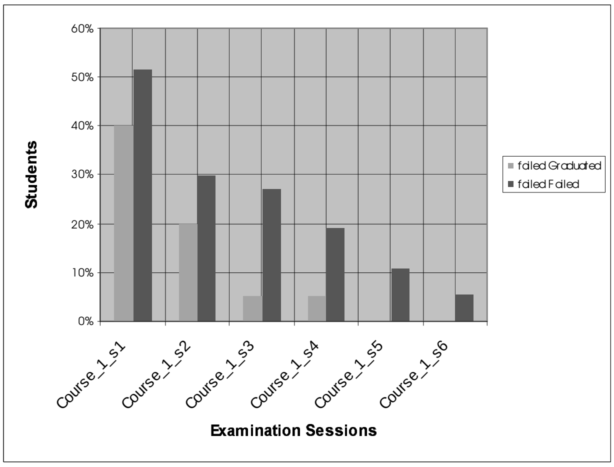

Figure 6. Course 1. Grade: failed

The students that did not present to the first examination session of course 1 have a high probability of not finishing the short degree (Figure 7).

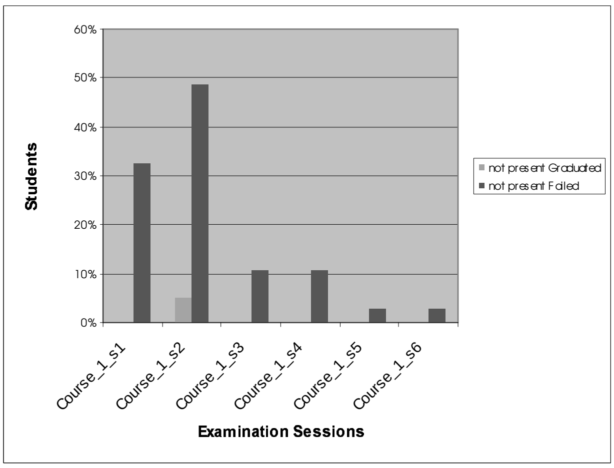

Figure 7. Course 1. Grade: not presented

In the last chart (Figure 8), we can see that about 27% of students that have not finished the short degree have presented only first year examination sessions.

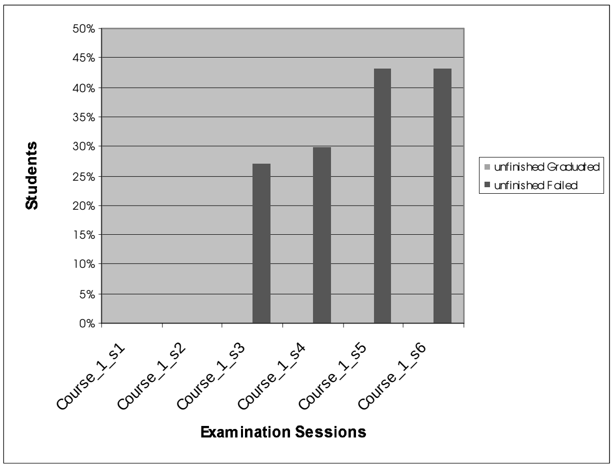

Figure 8. Course 1. Grade: unfinished

We have used the first 10 courses of the Curricular Plan because after our first results obtained through C4.5 we concluded that these courses have more influence in the students' academic behavior.

IV. CONCLUSIONS

In order to improve teaching activities within universities, we need powerful tools and techniques to analyze the relationships among all relevant data. In this paper, we have describe the combined use of two different machine learning techniques inside de KDD process to analyze the impact of the students relationships with some courses and the prediction of graduating or not.

V. FUTURE WORK

To extend our knowledge about the domain we will apply Inductive Logic Programming (ILP) tools, such as FOIL [7], to find more relationships among the attributes that could lead us to formulate some more general conclusions to increase the knowledge about the domain.

VI. REFERENCES

[1] W.J. Frawley, G. Piatetsky-Shapiro and C.J. Matheus, "Knowledge Discovery in Database: An Overview". In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Database, AAAI/MIT Press, 1991, 1-27.*

[2] R. Quinlan, "Induction of decision trees". In *Machine Learning Vol. 1, No.1*, pp 81-106. 1990.

[3] T. Mitchell, "Does Machine Learning Really Work?", In *AI Magazine*, Fall 1997.

[4] G. Piatetsky-Shapiro, "Knowledge Discovery in Real Databases" In *AI Magazine*, Winter 1991.

[5] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "Knowledge Discovery and Data Mining: Towards a Unifying Framework". In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996.*

[6] R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publisher, Inc, 1993.

[7] R. Quinlan, "Learning Logical Definitions from Relations". Machine Learning. Vol. 5, Number 3. pp. 239-266. 1990.

[8] R. Duda, and R. Hart, "Pattern Classification and Scene Analysis". John Wiley, New York. 1973.

[9] T. Mitchell, "Machine Learning", McGraw Hill..1997

[10] I. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann Publisher, Inc, 2000.

[11] C. Matheus, P. Chan and G. Piatetsky-Shapiro, "Systems for Knowledge Discovery in Databases" In TKDE special issue on *Learning & Discovery in Knowledge-Based Databases,* 1993.

[12] R. Engels, G. Lindner and R. Studer "A Guided Tour through the Data Mining Jungle" *In Proceedings of the 3th International Conference on Knowledge Discovery in Databases*, 1997.

[13] J. Han, Y. Cai and N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach", *In Proceedings of the 18th VLDB Conference*, 1992.

[14] R. Mooney homepage. www.cs.utexas.edu/users/mooney.